CS G154: JAVA PROGRAMMING 2

Item	Value
Curriculum Committee Approval Date	11/05/2024
Top Code	070720 - Database Design and Administration
Units	3 Total Units
Hours	90 Total Hours (Lecture Hours 36; Lab Hours 54)
Total Outside of Class Hours	0
Course Credit Status	Credit: Degree Applicable (D)
Material Fee	No
Basic Skills	Not Basic Skills (N)
Repeatable	No
Open Entry/Open Exit	No
Grading Policy	Standard Letter (S)

Course Description

Formerly: Data Structures with Java. This course covers advanced programming techniques and object oriented programming (OOP) in Java. Students will gain theoretical and hands-on experience with the implementation of typical data structures including arrays, queues, stacks, linked-lists, trees, hashing, and graphs used in programming applications. Principles of recursion, sorting, searching, optimization, classes, objects, inheritance, and polymorphism will be explored and practiced. PREREQUISITE: CS G153 or CS G175. Transfer Credit: CSU; UC. C-ID: COMP 132. **C-ID:** COMP 132.

Course Level Student Learning Outcome(s)

1. Course outcomes:
2. Apply complex data storage mechanisms and manipulation algorithms.
3. Develop programs that use abstract data structures.
4. Implement object-oriented class hierarchy and inheritance.

Course Objectives

- 1. Summarize advanced object-oriented programming concepts in Java.
- 2. Utilize modules including containers from the Java Application Programming Interface (API).
- 3. Create iterators and recursive algorithms to find, remove, reverse, and edit elements in containers.
- 4. Manipulate bitwise algorithms.
- 5. Apply exception handling.
- 6. Apply software development methodologies and debugging techniques.
- 7. Improve algorithm optimization and efficiency.
- 8. Design abstract data structures using classes and objects.
- 9. Apply inheritance, polymorphism, searching, and sorting.
- 10. Utilize stacks, queues, trees, and linked-lists.

Lecture Content

Mathematics review Algorithm analysis Running time calculations Solutions for the maximum subsequence sum problem Amortized analysis Logarithms in the running time Lists, Stacks, and Queues Abstract Data Types (ADTs) The list ADT Linked lists Lists in the Java collections API Iterators Nested and inner classes The stack ADT Applications The queue ADT Priority queues Circular queues Applications of queues Trees Implementation of a tree data structure Tree traversals Binary Search Trees (BST) AVL (Adel'son-Vel'skii and Landis) trees Tree balancing operations Splay trees B-Trees Sets and Maps in the API Sets Maps Implementation of TreeSet and TreeMap Hashing Hash function Separate chaining Hash tables Linear probing Quadratic probing Double hashing Rehashing Hash Tables in the API Priority Queues (Heaps) Binary heap Applications of priority queues The selection problem Event simulation Priority queues in the API Sorting Insertion sort Shellsort Heapsort Mergesort Quicksort Decision trees The disjoint set class Equivalence relations The dynamic equivalence problem Disjoint set data structure Graph algorithms Representation of graphs Topological sort Shortest-path algorithms Unweighted shortest paths Dijkstra's algorithm Acyclic graphs N etwork flow problems Minimum Spanning Tree (MST) Prim's algorithm Kruskal's algorithm Applications of Depth-First Search (DFS) Applications of Breadth-First Search (BFS) Undirected graphs Biconnectivity Euler circuits Directed graphs Algorithm Design Greedy algorithms Simple scheduling problem Huffman codes Approximate bin packing Divide and conquer algorithm Randomized algorithms Random number generators Skip lists Backtracking

Lab Content

Expressions, branches, loops, functions, and classes References and memory allocation Classes and object-oriented programming Inheritance and polymorphism Generics and parameterized types Linked lists and iterators Recursive techniques Stacks using arrays or linked lists Queues using arrays or linked lists Binary trees and tree balancing algorithms Hashing techniques Searching and sorting Graphs and Minimum Spanning Trees (MST) Create a project with the correct file structure Break the programs into appropriate classes Design a simple user interface to satisfy the user interactions

Method(s) of Instruction

- Lecture (02)
- DE Live Online Lecture (02S)
- DE Online Lecture (02X)
- Lab (04)
- DE Live Online Lab (04S)
- DE Online Lab (04X)

Reading Assignments

Textbook and websitesInstructor prepared materials

Writing Assignments

Students will be required to complete software development projects presented to them in the form of business automation problems requiring solution implementation. Students will be required to write documentation on their projects.

Out-of-class Assignments

An optional library research paper will promote further study and research in current Java programming or other related topics selected by the student and approved by the instructor.

Demonstration of Critical Thinking

Students will analyze requirements and select the appropriate programming structures for an efficient solution implementation. Testing and debugging will require students to perform data tracing and error isolation during program execution.

Required Writing, Problem Solving, Skills Demonstration

Students will be required to complete software development projects presented to them in the form of business automation problems requiring solution implementation. Students will be required to write documentation for their projects.

Eligible Disciplines

Computer science: Master's degree in computer science or computer engineering OR bachelor's degree in either of the above AND master's degree in mathematics, cybernetics, business administration, accounting or engineering OR bachelor's degree in engineering AND master's degree in cybernetics, engineering mathematics, or business administration OR bachelor's degree in mathematics AND master's degree in cybernetics, engineering mathematics, or business administration OR bachelor's degree in any of the above AND a master's degree in information science, computer information systems, or information systems OR the equivalent. Note: Courses in the use of computer programs for application to a particular discipline may be classified, for the minimum qualification purposes, under the discipline of the application. Master's degree required.

Textbooks Resources

1. Required Weiss, Mark Allen. Data Structures and Algorithm Analysis in Java, 3rd ed. Pearson (Latest), 2012 , ISBN: 0321322134. Rationale: .
2. Required Linag, Daniel Y. Introduction to Java Programming and Data Structures, Comprehensive, 12th ed. Pearson, 2020

Other Resources

1. Instructor prepared materials.